Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 20, Iss. 1, pp. 4-16, 2022

Comparison Between Different Architectures of Multilayer Perceptron Networks for Blocking Rate Prediction in Mobile Phone Networks

Gabriel da Silva Melo
Bruno Andre Santos , **Rogerio Martins Gomes**
Federal Center of Technological Education of Minas Gerais
gabrielsilvamelo96@gmail.com, {bsantos,rogerio}@cefetmg.br

Abstract – Blocking in mobile phone networks is a problem that consists of the refusal of the connection between a telephone device and a cell responsible for emitting the signal. The occurrence of blocking can indicate that a cell is close to congestion, leading to financial losses for telephone companies. This work developed three prediction systems using Multilayer Perceptron neural networks. Each system was modeled following different strategies: Direct, Recursive, and Direct Recursive, respectively. The training and test of the networks were carried out by using real data containing the history of blocking rates from a network of cells. The development stages consisted of analyzing the performance of each prediction system, varying the number of neurons in the hidden layers and the number of predicted steps from 1 (corresponding to 15 minutes ahead) to 20 (corresponding to 5 hours ahead). The system based on the Recursive strategy presented the lowest performance making predictions of short (15 minutes) and long (5 hours) terms with RMSE (Root Mean Squared Error) of approximately 13% and 40%, respectively, with a confidence interval between 27% and 29% considering all predictions. The systems based on the Direct and Direct Recursive strategies presented similar results, making predictions of short and long terms with RMSE of approximately 12% and 31%, respectively, with confidence intervals between 21% and 23% considering all predictions. Although the Direct and Direct Recursive systems obtained the lowest RMSE, the Direct Recursive is more advantageous as it requires fewer MLP networks. Consequently, it has simpler training and a lower computational cost.

Keywords – Blocking, Mobile Phone Networks, Time series analysis, Forecasting, Multi-step Prediction.

1 Introduction

Cell congestion in mobile phone networks is an important factor in ensuring signal quality. When a cell is congested, the communication of the devices is impaired. Several reasons can lead to congestion: power overloads, excessive user connections, excessive cell data traffic, and cell phone processor overload [1]. When a cell identifies that it is close to congestion, it can take some actions to mitigate the problem, such as: reducing the quality of voice calls, transferring connections to other cells, and blocking new connections [2]. However, these actions have drawbacks. Blocking connections and reducing the quality of voice calls, for instance, can damage the company's image and reduce the profit that could be obtained from customers using the network. Therefore, it is important to know, in advance, the intensity and duration of cell congestion to be able to carry out reactive measures that could avoid or minimize the negative effects.

Some works that use historical time series to predict congestion in mobile networks have been developed. Neto [3], for example, compared the performance of different models for the problem of predicting IP (Internet Protocol) data traffic in a mobile network. The models were developed using the following methods: Holt-Winters, Holt-Winters with double seasonality, TBATS, ARIMA (Autoregressive Integrated Moving Average), exponential smoothing methods, autoregressive neural network, RNN (Recurrent Neural Network), MLP, and LSTM (Long Short-Term Memory). Historical time series of IP data traffic was used to train the models. The models predicted the IP traffic one-step and multi-step ahead. The MLP network obtained the best results for the one-step predictions, and the LSTM the best results for multi-step predictions.

Dong, Fan, and Gu [4] carried out a comparison between ARIMA and the exponential smoothing model for predicting data traffic on a 4G network. They created four prediction scenarios, combining different configurations involving predictions for a single cell or a geographic region of cells and training models with weekday or weekend data. They showed that the ARIMA was better than the exponential smoothing model for predicting traffic on weekdays. On the other hand, the exponential smoothing model achieved better results than the ARIMA for predicting data traffic in a single cell.

In the study carried out by Torres et al. [5], a model was developed to predict the average download transfer rate in a 4G network. The database used to adjust the model parameters was obtained by probes attached to vehicles. The ARIMAX model (Autoregressive Integrated Moving Average with Explanatory Variable) was used to perform the prediction. The results showed that the prediction of the behavior of the network download transfer rate was feasible and with high accuracy.

Nikravesh et al. [6] analyzed the accuracy of network traffic prediction using the following machine learning techniques: MLP, MLP Weight Decay (MLPWD), and Support Vector Machines (SVM). The models were trained using data obtained from a single cell. Experiments were carried out by using pure data and by performing dimensionality reduction. The results showed that, for multidimensional data, the predictions with SVM obtained the best results. On the other hand, the MLPWD network model carried out the best predictions for one-dimensional data. Furthermore, predictions using multidimensional data had higher accuracy.

Chen et al. [7] implemented a model to predict network traffic using LSTM networks with GA (Genetic Algorithm). Initially, the LSTM networks were used to extract historical traffic data; then, the GA was used to identify appropriate hyper-parameters for the LSTM network. As a result, the developed model had a greater accuracy when compared to the prediction using techniques such as ARIMA and pure LSTM. In Wang, [8], a model combining LSTM networks with GPR (Gaussian Process Regression) to perform cell phone traffic prediction was proposed. The model surpassed predefined benchmark tests, especially for forecasting periods of congested traffic.

Ozovehe [9] compared different models to predict congestion in single cells of GSM/GPRS(2G) networks. Prediction models were created using MLP, RBF (Radial Basis Function), GMDH-PNN (Group Method of Data Handling-Probabilistic Neural Network), and ANFIS (Adaptive Neuro-Fuzzy Inference System) networks. These models were trained with data traffic and call success rates obtained over two years. GMDH-PNN networks obtained the best results.

Lima [10] implemented different models of neural networks performing a one-step prediction of blocking rate in three cells of a 3G network. The prediction was carried out based on the following five parameters reported by the cells: the blocking rate, the availability rate of the cell, the data traffic, the exchange rate of user antennas, and the voice traffic. The forecast methods implemented were: MLP, RBF, and LSTM networks. The LSTM networks obtained the best results.

While most works in the literature develop models that predict data traffic, the model developed here predicts the behavior of the time series of blocking rates from the moment a first blocking occurs. Forecasting the behavior of the time series blocking rates can be relevant in cases where the blocking rate has a long duration, as it allows acting on the network to improve the quality of service. Another important aspect of this work is that it aims to develop a generic forecasting model that can be used in different cells, which is obtained by adjusting the model's parameters using the history of blocking rates from several cells.

This work developed and compared the performance of three systems that perform blocking rate predictions. The difference between these systems lies in the strategy used to connect and train the neural networks of each model. The first system uses a Direct strategy, in which there is a model to predict each future state. The second system uses a Recursive strategy, in which a single network uses its predictions as inputs to make future predictions. The third system combines the Direct and Recursive strategies. It has a single model for each predicted future state and uses its predictions as input to make new predictions. The systems were trained and tested using a database containing information about the blocking rate of 3G networks from different regions of Brazil. The methodology used to develop the work, the results and conclusions are presented in the sections 2, 3 and 4, respectively.

2 Methodology

Initially, it is presented, in Sec. 2.1, how the blocking time series to train the models were obtained. Then, in Sec. 2.2, three systems that were developed using different strategies to perform multistep predictions are explained. Finally, in Secs. 2.3 and 2.4, the training and the evaluation methods for each system are described.

2.1 Database

The database was obtained from a 3G network consisting of 21 thousand cells for six months and with the following fields: 1) cell identification, 2) timestamp (day and time), and 3) blocking rate.

A preprocessing step was carried out to obtain the time series with blocking rates greater than zero. The blocking rate remains at 0 (zero) most of the time because this is the expected behavior of a cell, while the blocking is an anomaly. Therefore, time series with blocking rates greater than zero were present only in 1.26% of the database – these time series were used to train the models. To illustrate how this preprocessing step was carried out, consider the illustration shown in Figure 1. The information about the blocking rate is provided by each cell every 15 minutes. As the blocking rate, in the illustration, was greater than 0 within the intervals [00h00m,1h00m] and [1h45m, 3h30m], two time series were created – the first one lasting 1 hour and the second one lasting 1 hour and 45 minutes.

One of the characteristics observed in the database is that the length of the time series is short, as shown in Table 1. For example, 74.40% of the time that blocking is reported, it does not last longer than 15 minutes. Another characteristic observed is that the blocking rate of the first elements is concentrated in a small interval, while more distant elements are more dispersed, as shown in Figure 2. As it is explained in Sec. 2.3, it was necessary to balance this database to avoid overfitting of the models during the training step.

2.2 Multistep Forecasting Systems

Three types of multistep forecasting systems were developed using one of the following strategies: Direct, Recursive, and Direct Recursive (DirRec). [11–15].

2.2.1 Direct System

The Direct strategy uses one forecasting model to predict each step in the future. For example, considering the prediction of five steps in the future from a time series containing ten values, five different models should be built, each one with ten inputs. The

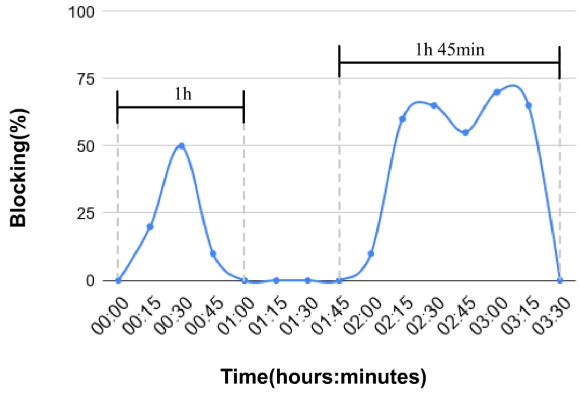


Figure 1: Illustration of how the blocking rate (y-axis) changes over time (x-axis). The blocking rate varies from 0 (no blocking) to 100%.

Table 1: Number of time series in the database grouped by the duration of the blocking rates.

Duration of blocking	Number of Series	Percentage
15min	1.624.138	74.40%
30min	273.920	12.54%
45min	94.280	4.30%
1h-6h	168.996	7.75%
6h-12h	11.988	0.54%
12h-24h	8.207	0.37%
Total	2.180.154	100%

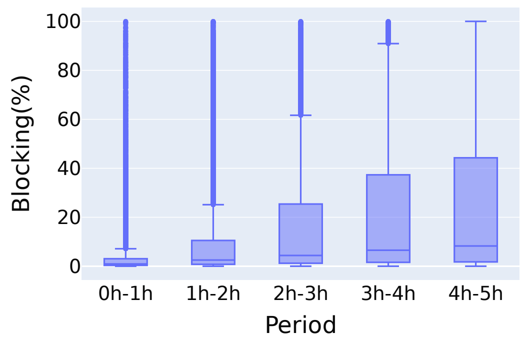


Figure 2: Distribution of blocking rates grouped by time intervals.

$1st$ model will predict the $11th$ step, the $2nd$ model will predict the $12th$ step, and so on until the prediction of the $15th$ step by the $5th$ model. Note that the models predict five steps in the future using the same ten inputs. The $5th$ model, for example, takes ten inputs and predicts only the $15th$ step without knowing the $11th$, $12th$, $13th$, and $14th$ steps. This model can be formally described as in Eq. (1).

$$\hat{y}_{t+h} = MLP_{M,h}(y_t, y_{t-1}, ..., y_{t-M+1}) \ with \ 1 \leqslant h \leqslant H \tag{1}$$

where M is the number of inputs and $(y_t, y_{t-1}, ..., y_{t-M+1})$ defines the vector of inputs. Each model was implemented as a MLP (Multi-Layer Perceptron) network and predicts h-steps ahead, i.e. for predicting all values from $\hat{y}_{t+1}$ to $\hat{y}_{t+H}$, H different models should be build.

Following this strategy, a system called *Direct System* was developed, containing 210 MLP networks. This system consists of 20 subsystems, each of which receives a different amount of inputs (from 1 to 20), as illustrated in Figure 3. Subsystem 1 receives one input (y_1) and predicts the next 20 steps using an MLP network for each step. The $\text{MLP}_{1.1}$ network receives the input (y_1) and predicts the next value (t+1) of the blocking rate ($\hat{y}_2$). The $\text{MLP}_{1.20}$, which also receives the input y_1, predicts the blocking rate twenty steps ahead ($\hat{y}_{21}$). Subsystem 2 takes two inputs (y_1, y_2) and predicts the next 19 steps (from $\hat{y}_3$ to $\hat{y}_{21}$).

The number of inputs in each subsystem varies until the last subsystem that receives 20 inputs and predicts the next step ($\hat{y}_{21}$). Twenty networks were developed for the first system, nineteen networks for the second one, eighteen networks for the third one, and so on, up to one network for the last system, totaling 210 MLP networks for the whole Direct System.

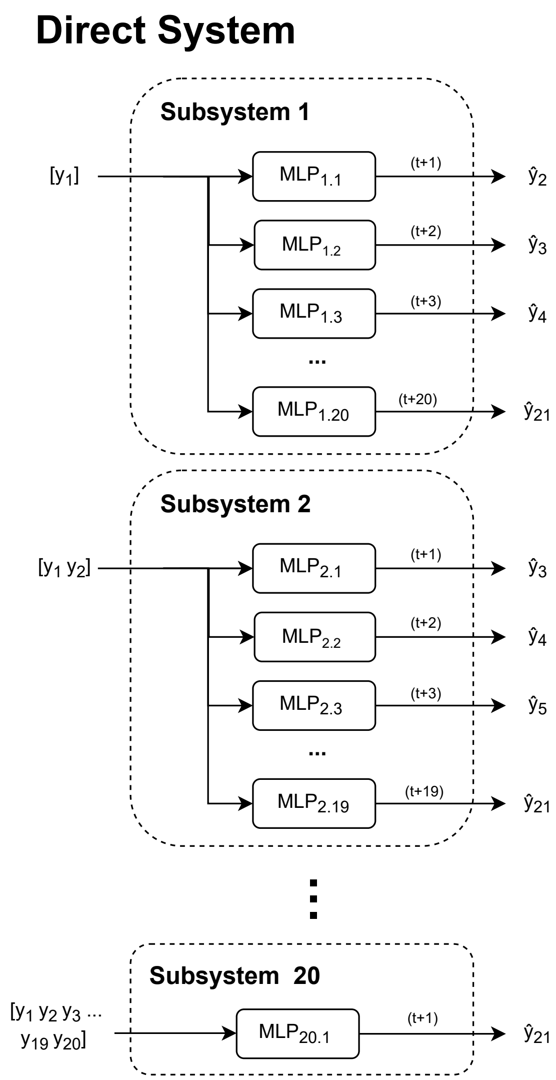


Figure 3: Direct forecasting system architecture. Each dotted-line rectangle represents a subsystem. Note that the number of inputs (e.g. [y_1 y_2]) varies for each subsystem. Each solid-line rectangle represents a different MLP network. In the sub-index $m.n$ of each MLP network, m represents the number of entries in the network, and the value of n represents the step predicted.

2.2.2 Recursive System

The recursive system was implemented using one MLP network with 20 inputs and one output, as formally described in Eq. (2).

$$\hat{y}_{t+1} = MLP_r(y_t, y_{t-1}, ..., y_{t-M+1}) \tag{2}$$

where M is the number of inputs and $(y_t, y_{t-1}, ..., y_{t-M+1})$ defines the vector of inputs. As the number of inputs is fixed (M=20) and the times series of blocking rates have a different number of elements (from 1 to 20, depending on their duration), time series smaller than 20 had their missing values completed with zeros (0). The same model (MLP_r) is used to predict the next value ($\hat{y}_{t+2}$), as shown in Eq. (3).

$$\hat{y}_{t+2} = MLP_r(\hat{y}_{t+1}, y_t, y_{t-1}, ..., y_{t-M+2}) \tag{3}$$

Note that the predicted value $\hat{y}_{t+1}$ is used as input to predict $\hat{y}_{t+2}$. To predict H-steps ahead, the values from $\hat{y}_{t+2}$ to $\hat{y}_{t+H}$ are predicted iteratively. For example, considering the prediction of five steps in the future from a time series containing ten values. Firstly, the time series is completed with zeros as the model has 20 inputs. Then the prediction of the $\hat{y}_{11}$ is carried out. The value of $\hat{y}_{11}$ is concatenated to the time series of 10 elements, generating a new time series with 11 elements. Then, the prediction of $\hat{y}_{12}$ is carried out using the time series with 11 elements as input. This process is repeated until the prediction of the $\hat{y}_{15}$ (with the first ten elements belonging to the real time series and the other five elements predicted by the system). The recursive system is illustrated in Fig. 4.

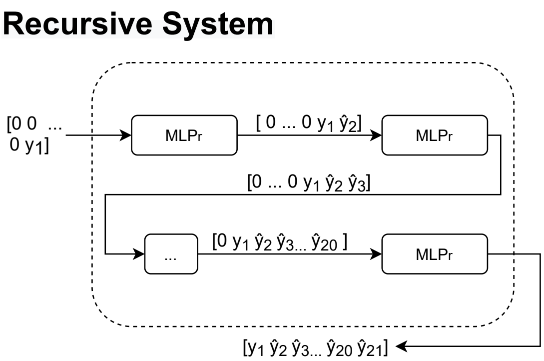


Figure 4: Recursive forecasting system. This system consists of one MLP network that receives 20 inputs and generates one output. The predicted output ($\hat{y}_n$) is applied to the network recursively.

2.2.3 Direct Recursive System

The Direct Recursive System (DirRec) has 20 MLP networks with different input sizes (from 1 to 20). All networks predict only the next step (t+1). This system is modeled by using twenty $MLP_{M,1}$ networks, as described in Eq. (1), with $1 \leqslant M \leqslant 20$. While in the Direct System the vector of inputs consists of real values, in the DirRec system, the vector of input has real and predicted values, as it is illustrated in Fig. 5. When the time series has one value, the input y_1 is applied to MLP$_{1.1}$ to generate the prediction of step 2 ($\hat{y}_2$). The values y_1 and $\hat{y}_2$ are used as input to the MLP$_{2.1}$. The MLP$_{2.1}$ network, in turn, predicts $\hat{y}_3$, which is concatenated with the previous values [y_1 $\hat{y}_2$] generating the input [y_1 $\hat{y}_2$, $\hat{y}_3$] to MLP$_{3.1}$. This process is repeated until the last network (MLP$_{20.1}$) that predicts $\hat{y}_{21}$. When the time series has two real values [y_1 y_2], these values are applied directly to MLP$_{2.1}$. The same rationale is applied for all input sizes until 20. In total, the Direct Recursive System has 20 networks.

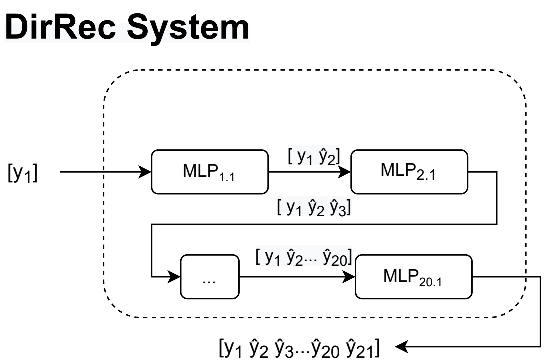


Figure 5: DirRec Prediction System. Solid-line rectangle represents a different MLP network. In the sub-index $m.n$ (of each MLP network), the value of m represents the number of inputs in the network, and the value of n represents the predicted step. The networks have different input sizes (from 1 to 20) and predict the step t+1.

It is important to note that the prediction systems proposed in this work aim to predict how the blocking rate evolves from the moment a first blocking has been reported. That is why each system was modeled to predict the blocking rate based on different input sizes (from 1 to 20).

2.3 Training the MLP networks

The MLP networks were implemented using the Tensor Flow [16]. The models were implemented using the 'RMSE' for the loss function, 'Adam' for the optimization function, and 'ReLU' for the activation function. The networks were trained using the K-Fold cross-validation method with K equals four and with 100 epochs [17].

The definition of the number of neurons in the hidden layers of the MLP networks was carried out by analyzing the prediction error using 5, 10, 20, 30 neurons in the first layer and 0, 5, 10, 20, and 30 in the second layer. The prediction errors of the t+1 step were calculated for models with different numbers of inputs, from 1 to 20. Finally, the configuration that obtained the smallest error was chosen to implement the systems.

The networks were trained with a balanced database of time series was created. The balancing was necessary as most of the blocking rates were less than 40%, which could cause overfitting of the models. The balancing was done by creating four sets;

in each set, the value that had to be predicted by the network was within a specific range. The $1st$ part of the balanced database consisted of time series whose predicted value was within [0, 25]. The $2nd$, the $3rd$ and the $4th$ parts consisted of predicted values within [25, 50], [50, 75] and, [75, 100], respectively. Figure 6 illustrates how the balancing was carried out. Each MLP network of the Direct and DirRec system was trained with 2000 time series (500 in each interval). The MLP network of the Recursive system was trained with 8000 times series. The number of elements in each database was defined by analyzing the convergence of the error during the training.

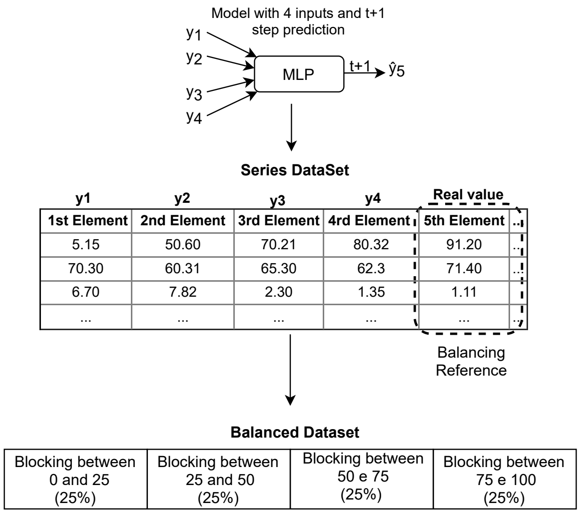


Figure 6: Illustration of how the database was balanced for training the networks. A schematic representation of the network with 4 inputs (y_1, y_2, y_3, y_4) and one output ($\hat{y}_5$) is shown at the top. The table in the middle shows a database consisting of time series with five elements each (4 elements correspond to y_1, y_2, y_3, y_4, respectively, and the fifth element corresponds to the value that has to be predicted by the network). Twenty-five percent of the database has the fifth element within [0, 25), 25% within [25, 50), 25% within[50, 75), and 25% within [75, 100], as illustrated at the bottom.

2.4 Evaluation the Forecasting Systems

After training each MLP network and implementing the architecture of the forecasting systems (as described in Section 2.2), the evaluation of the Direct, Recursive, and DirRec systems was carried out. The database for evaluating the systems consisted of 2,000 time series. These time series were selected based on their $21st$ element (which is the last value predicted by the systems). Twenty-five percent (25%) of the time series had the $21st$ element between 0 and 25, 25% between 25 and 50, 25% between 50 and 75, and 25% between 75 and 100.

The evaluation process for each system was carried out in 20 steps. In the first step, only the $1st$ element of the time series was provided to the Direct, Recursive and DirRec systems. From the $1st$ element, predictions were carried out up to the $21st$ element. In the second step, the first two elements of the time series were provided to the systems, and the following nineteen elements were predicted. This process repeated until the last step, in which 20 elements of the series were provided to the systems and the $21st$ element were predicted. For each step, the RMSE was calculated.

Another system, called Naive, was also implemented to perform the predictions providing a baseline of comparison for the other three systems. The following methods were tested to implement the Naive system: 2nd, 3rd, and 4th-degree polynomial regression and the mean of the input values. The method that obtained the best result (the smallest RMSE) was the mean of the inputs. For this reason, it was chosen to be used as the baseline of comparison for the systems.

3 Results

3.1 Defining the number neurons and layers in the MLP networks

The number of neurons in the hidden layers of the MLP networks was defined by training 1,600 networks. This amount of networks is due to the 20 possible input sizes, multiplied by 20 combinations of neurons in the hidden layers, and 4 training combinations (due to K-Folds).

Tables 2 and 3 show the RMSE of the training step varying the number of layers of networks with 5 and 10 inputs, respectively. Note that different combinations of neurons in the layers generate small variations in the RMSE. For the network with five inputs, the mean RMSE considering all combinations of neurons in each layer is 15.535 with a standard deviation of 0.201. For the network with ten inputs, the mean RMSE considering all combinations of neurons in each layer is 13.392 with a standard deviation of 0.104. Minor variations in the RMSE repeated for all input sizes, as shown by the standard deviation in Table 4. It is observed that the error decreases with the increase in the number of inputs, stabilizing around 13 for inputs greater than or equal to 8 elements.

Table 2: RMSE for different numbers of neurons in the network that has 5 inputs. The rows represent the number of neurons in the first hidden layer (L1), and the columns the number of neurons in the second layer (L2).

L1xL2	**0**	**5**	**10**	**20**	**30**
5	15.982	15.795	15.733	15.733	15.526
10	15.890	15.643	15.533	15.472	15.324
20	15.635	15.445	15.404	15.333	15.411
30	15.398	15.445	15.312	15.334	15.355

Table 3: RMSE for different numbers of neurons in the network that has 10 inputs.

L1xL2	**0**	**5**	**10**	**20**	**30**
5	13.587	13.540	13.487	13.333	13.441
10	13.524	13.352	13.404	13.308	13.312
20	13.443	13.513	13.403	13.347	13.392
30	13.221	13.232	13.415	13.276	13.300

Table 4: Mean and Standard Deviation of tests with different numbers of neurons grouped by the input size.

Input size	**Mean RMSE**	**Std(RMSE)**
1	22.826	0.625
2	18.296	0.436
3	16.668	0.339
4	15.790	0.280
5	15.535	0.201
6	14.739	0.184
7	14.698	0.138
8	13.887	0.126
9	13.683	0.147
10	13.392	0.104
11	13.556	0.070
12	13.217	0.053
13	12.926	0.078
14	13.196	0.087
15	13.008	0.117
16	13.088	0.108
17	12.964	0.084
18	13.390	0.069
19	13.369	0.075
20	13.623	0.048

Table 5: Mean RMSE grouped by the number of neurons in each layer. Each cell contains the mean RMSE considering 20 input sizes.

L1xL2	0	5	10	20	30
5	14.955	14.780	14.658	14.541	14.534
10	14.753	14.672	14.551	14.505	14.495
20	14.631	14.577	14.505	14.513	14.504
30	14.549	14.494	14.508	14.552	14.573

Table 6: RMSE of MLPs networks with 30 neurons in the first hidden layer and 5 neurons in the second hidden layer. Each element of the table shows the mean RMSE considering four tests generated by the K-Folds. Rows represent the number of inputs, and the columns represent the step in the future predicted by the model.

Input size	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20
1	22.7	27.2	26.4	28.3	30.0	27.5	29.8	29.0	29.7	28.7	28.5	29.0	30.2	29.3	29.2	29.1	29.0	29.0	29.1	29.2
2	18.4	22.7	24.5	24.2	25.1	27.1	26.7	27.3	27.2	28.1	27.7	27.8	28.1	28.2	28.1	28.0	28.0	28.0	28.0	
3	16.5	20.8	22.9	23.1	24.3	25.5	25.5	26.5	26.1	26.2	27.4	26.9	26.9	27.1	27.4	28.1	27.4	28.1		
4	15.6	19.4	21.3	22.1	23.1	23.9	24.2	25.5	24.6	25.8	27.4	26.6	26.8	27.1	27.4	26.9	26.9			
5	15.4	18.8	19.8	22.0	21.6	23.6	24.2	24.9	24.7	24.7	26.4	26.3	26.3	25.5	26.5	26.9				
6	14.5	17.4	19.3	20.2	20.7	22.4	23.0	23.7	23.9	25.1	25.4	25.2	25.7	25.9	26.3					
7	14.6	17.1	18.5	19.4	19.7	21.9	22.4	23.7	24.6	23.8	24.8	26.1	24.5	25.3						
8	13.7	16.3	17.8	19.2	19.6	21.0	21.6	22.4	22.9	24.2	23.7	24.4	24.2							
9	13.5	16.3	17.6	18.4	19.7	20.7	21.0	22.2	23.4	23.8	22.6	23.6								
10	13.2	15.6	17.0	17.9	19.1	20.3	21.3	22.3	22.7	23.1	22.9									
11	13.4	15.5	17.3	17.8	19.6	20.8	20.8	21.0	21.5	23.0										
12	13.2	16.0	17.2	17.7	19.6	19.4	20.7	20.7	21.9											
13	13.0	16.3	17.5	18.6	18.7	19.5	19.7	20.6												
14	13.2	15.5	16.9	18.0	18.8	19.9	19.8													
15	13.0	16.2	17.1	18.3	19.6	19.9														
16	13.0	15.5	16.7	18.3	19.4															
17	12.9	15.6	16.8	18.4																
18	13.4	15.4	17.0																	
19	13.4	15.3																		
20	13.6																			

Table 5 shows the mean RMSE for all networks grouped by the number of neurons in each layer. For instance, using five neurons in the first layer and none in the second layer, the mean RMSE considering all input sizes (from 1 to 20) is 14.955. Based on the results shown in this table, the MLPs of the forecasting systems were implemented by using 30 neurons in L1 and five neurons in L2, which is the configuration that obtained the lowest mean RMSE. After defining the number of layers and neurons, the MLPs were trained to predict the blocking rate for N steps ahead, as described in the next section.

3.2 Training the MLP networks

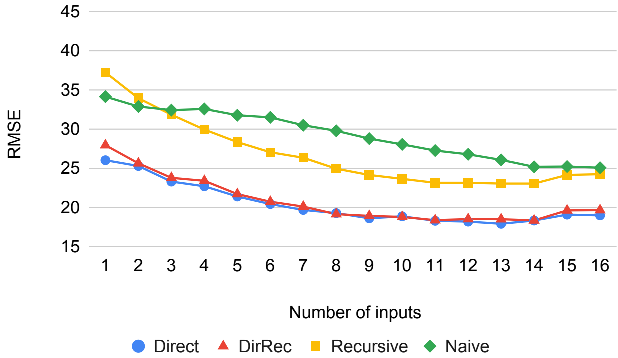
In total, 210 MLP networks were trained to predict the blocking rate for each input size (from 1 to 20) and each prediction step (from 1 to 20). The results are shown in Table 6. For five inputs, for example, the RMSE is 15.4 for predicting one step (t+1) and 26.9 for sixteen steps (t+16). All these 210 networks were used to implement the Direct system, as explained in Sec. 2.2.1 and illustrated in Fig. 3.

While the Direct system requires a different network for different input sizes, the Recursive system has only one network for all input sizes and to predict all steps in the future, as explained in Sec. 2.2. The mean RMSE obtained after training the MLP network of the Recursive system was 15.56 with a standard deviation of 0.33.

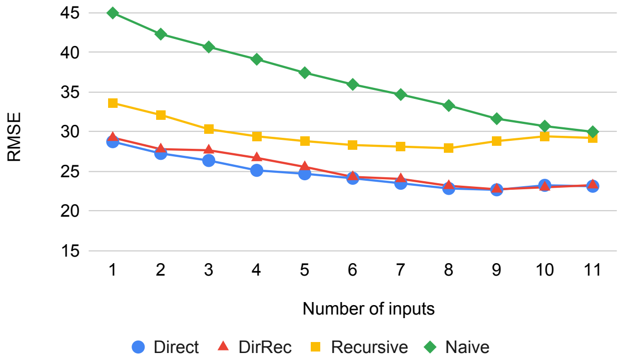
The architecture of the DirRec system was implemented by using the 20 networks that predict the step t+1, which RMSE are shown in Table 6. For example, the RMSE is 22.7 when the network predicts the step t+1 by using one input; and the RMSE is 13.6 when the network predicts the step t+1 by using twenty inputs, as shown in Table 6. Note that the Direct and DirRec systems share the same networks that predict the step t+1. However, these systems differ in how they predict the steps t+N for N greater than 1, as explained in Sec. 2.2.

3.3 Analyzing the predictions of the systems

The performance of the systems predicting the steps t+5 and t+10 is shown in Figure 7. Figure 7a shows that the error of the Direct system using only one input is approximately 26. In this case, the prediction of the $6th$ element (t+5) is made by using the $1st$ element of the time series. In Figure 7b, the predictions of step t+10 are shown. The error of the Direct system using only one input is approximately 29. In this case, only the $1st$ element of the time series was used to predict the $11th$ element. The errors of the Direct and DirRec systems stabilize around the same value. It can also be seen that the error of the Naive system is always greater than the error of the Direct and DirRec systems.



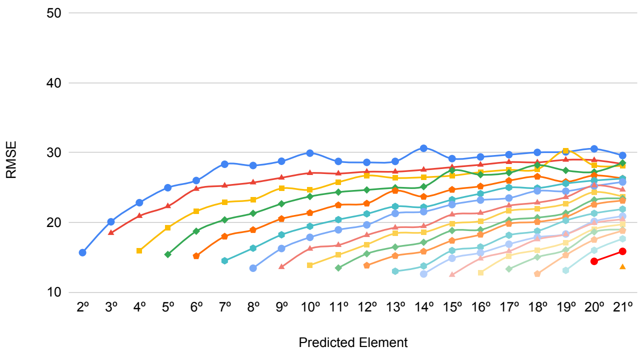
(a) RMSE of the step t+5 using the Direct, Recursive, DirRec, and Naive systems (see legend).



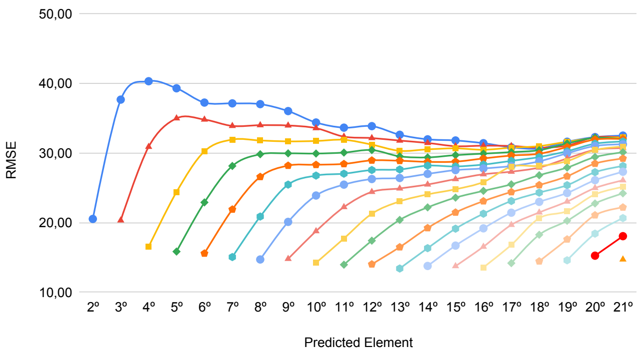
(b) RMSE of the step t+10 using the Direct, Recursive, DirRec, and Naive systems (see legend).

Figure 7: RMSE of the forecast for steps t+5 and t+10.

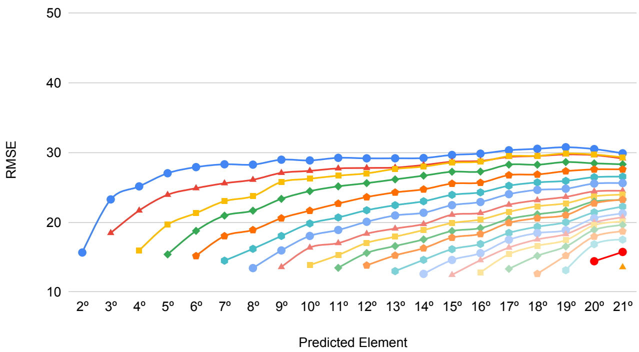
In Figure 8, the RMSE of the predictions of all elements (up to the 21st) for all input sizes (from 1 to 20) is presented. In Figure 8a, the curve labeled "2 Inputs" (see legend) means that the Direct system is making the predictions using two real values of blocking rates. The first prediction made by the system is the $3rd$ element. The following values of the curve are the errors of predictions for the following elements (up to the 21st element).



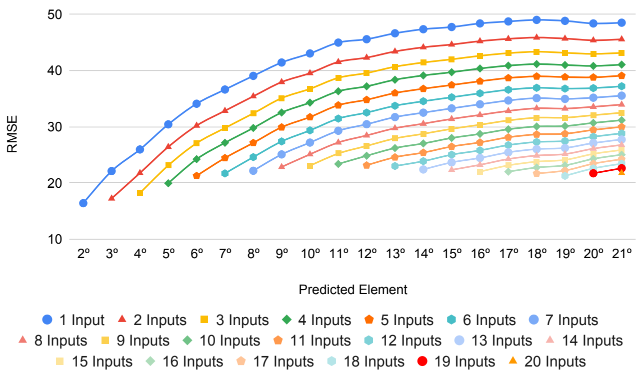
(a) RMSE of the Direct system.



(b) RMSE of the Recursive system.



(c) RMSE of the DirRec system.



(d) RMSE Naive system.

Figure 8: RMSE of each system. The horizontal axis shows the element predicted, and the vertical axis shows the RMSE error of the prediction. The curves (see legend) show the RMSE of the predictions for different input sizes.

The systems were compared by analyzing the distribution of RMSEs considering all possible inputs (from 1 to 20) and all prediction steps. As shown in Figure 9, the Direct and DirRec systems obtained similar performances. The confidence interval for the Direct and DirRec systems was from 21.13% to 22.86% and from 21.02% to 22.97%, respectively, with the same median of 22%. For the Recursive system, the confidence interval was from 27.15% to 29.03% with a median of 28.09%, which indicates a disadvantage in using this system compared to the Direct and DirRec systems. For the Naive system, the confidence interval was from 29.09% to 31.90% with a median of 30.50%, which suggests an advantage in using the systems to predict the behavior of the blocking rates. We point out that the purpose of the Naive system is to provide a baseline of comparison.

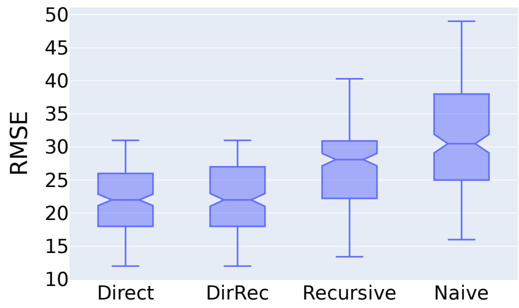
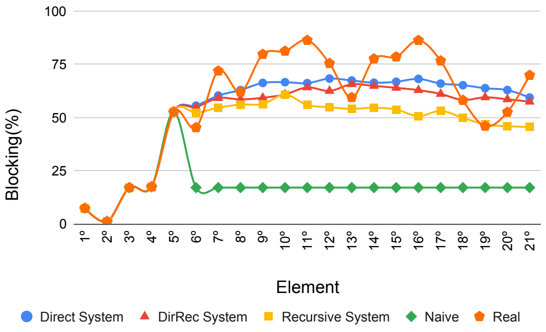


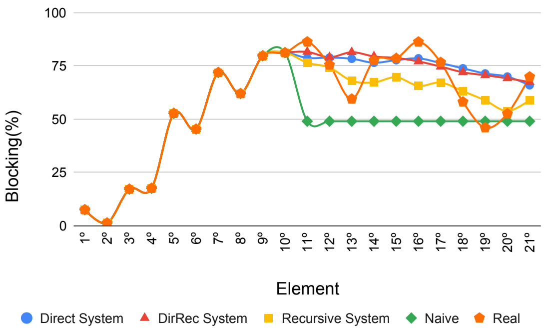
Figure 9: Comparison between the RMSE distributions (y-axis) of the Direct, Recursive, DirRec and Naive (x-axis) systems.

3.4 Example of predictions made by the systems

In Figure 10, the application of the models for the prediction of a time series is presented. The Direct and DirRec models generate similar predictions. In Figure 10a, the prediction of the $6th$ element (t+1) made by the Direct and DirRec systems was close to the value of the real blocking rate: the real value was 16.54, and the value predicted by both systems was 18 .85. Note that the predictions of DirRec and Direct are always the same for the prediction of step t+1. That is because the network that predicts the first step is the same for both systems. In the prediction of the $7th$ element (t+2), the Direct and DirRec systems also predicted results close to the real value: the real value was 18.19, the Direct system predicted 19.64, and the DirRec system predicted 18.19. Although the predictions after step t+2 do not capture the fluctuations in the blocking rate, they follow the blocking rate tendency. The blocking rates predicted by the Recursive system are lower than the values predicted by the Direct and DirRec systems. The Naive model underestimated the blocking percentage throughout the forecast window for both input sizes.



(a) Prediction of the blocking rate from $6th$ to $21st$ elements of the time series using 5 input values.



(b) Prediction of the blocking rate from $11th$ to $21st$ elements of the time series using 10 input values.

Figure 10: Blocking rate prediction cases. The orange curve shows the real time series (see legend). The real time series is the same in graphics (a) and (b). While in (a) the prediction uses the first 5 elements of the time series, in (b) it uses the first 10 elements.

4 Conclusions

The present work studied the performance of different MLP network architectures to predict blocking rates in mobile networks. The predicting systems were modeled following three strategies: Direct, Recursive, and Recursive Direct. The systems networks were evaluated considering predictions from one step ahead (15 minutes of blocking) to 20 steps ahead (5 hours of blocking).

The recursive system, the simplest system in terms of implementation, obtained the lowest performance for short (15 minutes) and long-term (5 hours) predictions. Evaluating the results and observing the predictions made by the Direct and DirRec systems, we see that both systems make similar predictions and have similar errors. In the short and long-term predictions, both systems obtained an RMSE of approximately 12% and 31%, respectively. Furthermore, the distributions of the RMSEs of the two systems were also similar (same median and close confidence intervals). For these reasons, in terms of the quality of predictions, there is no advantage of one system over the other. However, the use of the DirRec System is more viable because it requires a smaller number of networks (20 networks for the DirRec system and 210 for the Direct). In this way, the DirRec has simpler training and a lower computational cost.

In future works, the predictions could be made using a database with other features related to the blocking rate, such as the volume of packet traffic and voice traffic. Using more features could reduce the error of predictions and allow predicting the onset of a lock (and not just predicting how it behaves after starting). The systems could also be implemented with other prediction models, such as ARIMA, LSTM, and Holt-Winters. In addition to looking for ways to improve the model's performance, another line of research would be studying blocking prediction for other network technologies, such as 4G.

Finally, it is expected that this work can be used as a reference for developing other models to predict the blocking rate, allowing the creation of a functional tool for mobile phone network monitoring systems.

Acknowledgements

The authors would like to thank CEFET-MG for its financial support, without which this work would not have been possible.

List of Acronyms

ARIMA - Autoregressive Integrated Moving Average.
DirRec - Direct Recursive system.
LSTM - Long Short-Term Memory.

MLP - Multi-Layer Perceptron.
RBF - Radial Basis Function.
RMSE - Root Mean Squared.

References

[1] S. Kyriazakos, N. Papaoulakis, D. Nikitopoulos, E. Gkroustiotis, C. Kechagias, C. Karambalis, and G. Karetsos, "A comprehensive study and performance evaluation of operational gsm and gprs systems under varying traffic conditions," in *11th IST Mobile Summit*, Citeseer, 2002.

[2] E. B. Rodrigues, F. R. Cavalcanti, and S. Wänstedt, "Congestion control for wireless cellular systems with applications to umts," in *Optimizing Wireless Communication Systems*, pp. 141–185, Springer, 2009.

[3] C. L. Neto, "Previsão de Tráfego em Redes IP," 2018. Monografia (Bacharel em Engenharia Elétrica), UFMG (Universidade Federal de Minas Gerais), Belo Horizonte, Brasil.

[4] X. Dong, W. Fan, and J. Gu, "Predicting lte throughput using traffic time series," *ZTE Communications*, vol. 13, no. 4, pp. 61–64, 2015.

[5] P. Torres, P. Marques, H. Marques, R. Dionísio, T. Alves, L. Pereira, and J. Ribeiro, "Data analytics for forecasting cell congestion on LTE networks," in *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–6, IEEE, 2017.

[6] A. Y. Nikravesh, S. A. Ajila, C.-H. Lung, and W. Ding, "Mobile network traffic prediction using mlp, mlpwd, and svm," in *2016 IEEE International Congress on Big Data (BigData Congress)*, pp. 402–409, IEEE, 2016.

[7] J. Chen, H. Xing, H. Yang, and L. Xu, "Network traffic prediction based on lstm networks with genetic algorithm," in *International Conference on Signal and Information Processing, Networking and Computers*, pp. 411–419, Springer, 2018.

[8] W. Wang, C. Zhou, H. He, W. Wu, W. Zhuang, and X. S. Shen, "Cellular traffic load prediction with lstm and gaussian process regression," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2020.

[9] A. Ozovehe, O. U. Okereke, A. E. Chibuzo, and A. U. Usman, "Comparative analysis of traffic congestion prediction models for cellular mobile macrocells," *European Journal of Engineering Research and Science*, vol. 3, no. 6, pp. 32–38, 2018.

[10] G. P. Lima, "Previsão de bloqueio em células 3G - Um estudo de caso," 2018. Monografia (Bacharel em Engenharia Elétrica), UFMG (Universidade Federal de Minas Gerais), Belo Horizonte, Brasil.

[11] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16-18, pp. 2861–2869, 2007.

[12] A. S. Weigend, *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.

[13] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *European business intelligence summer school*, pp. 62–77, Springer, 2012.

[14] S. B. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert systems with applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[15] A. Sorjamaa and A. Lendasse, "Time series prediction using DirRec strategy," in *Esann*, vol. 6, pp. 143–148, 2006.

[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[17] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*, pp. 532–538. Boston, MA: Springer US, 2009.